

Claims

1. (Amended) A method of allocating resources including scheduling jobs from among a plurality of resources of a work-producing system, said method comprising the steps of:
 - (a) sorting, in a predetermined order, available resources by last task assignment, a number of tasks performable, rate per task, and cost per hour, and
5 determining at least one queue responsive to said sorting, wherein the rate per task characterizes the units processed in the task for a time period;
 - (b) normalizing the at least one queue by dividing a current task queue by an average rate of the available resources for each task in the current task queue; and
10 (c) assigning the available resources to at least one task with a predetermined normalized queue subject to at least one task constraint.
2. A method of allocating resources according to claim 1, further comprising the step of redetermining the at least one queue after assignment of the available resources, and designating the assigned resource unavailable until a predetermined time when the assigned available resources expires.
3. A method of allocating resources according to claim 1, further comprising the step of incrementing time to time of a next event.

4. A method of allocating resources according to claim 1, wherein the at least one task constraint includes maximum resource capacity, defined start and end times, and scheduled down time.

5. A method of allocating resources according to claim 1, wherein the at least one task constraint includes at least one team assignment constraint, and the available resources are assigned to the at least one task until the at least one team assignment constraint is satisfied.

6. A method of allocating resources according to claim 1, wherein said assigning step (c), further comprises the steps of assigning the available resources to the at least one task for a maximum time of task, and removing the at least one task from a resource skill set.

7. A method of allocating resources according to claim 1, wherein the at least one constraint includes an end of shift constraint, and wherein the available resources are not assigned to the at least one task when the assignment violates the end of shift constraint.

8. A method of allocating resources according to claim 1, wherein the predetermined order comprises an ascending order.

9. A method of allocating resources according to claim 1, wherein the predetermined normalized queue comprises a largest normalized queue.

10. A method of allocating resources according to claim 1, wherein said sorting step (a), said normalizing step (b) and said assigning step (c) are performed according to a resource allocation model, and

wherein the resource allocation model includes entities with variable attributes
5 having variable quantities that transform through at least one network of nodes.

11. A method of allocating resources according to claim 10, wherein each node of the at least one network of nodes includes an associated set of attributes and parameters.

12. A method of allocating resources according to claim 11, wherein the attributes are qualitatively defined through at least one of nominal, graphical and symbolic conventions.

13. A method of allocating resources according to claim 12, wherein the available resources include the attributes of the nodes, and the available resources undergo transformational processes arriving at least one arbitrary state or passing through a series of states that may become the attributes of the resources.

14. A method of allocating resources according to claim 11, wherein the parameters are specified as at least one of inputs, outputs, capacities, operational processes, functional behaviors, movement logics, and other dynamic parameters.

15. A method of allocating resources according to claim 10, wherein the resource allocation model stores at least one of historical values, theoretical values, the attributes and constellations of the nodes, and wherein the resource allocation model provides multiple bases of comparison for monitoring, measuring, and evaluating real-
5 time operational data and operational performance for management functions.

16. A method of allocating resources according to claim 10, wherein the resource allocation model includes significance and performance criteria, associated tableaux and scenarios, and wherein abstract model elements are stored as at least one of the parameters and the attributes, and as at least one of functional, logical, graphical and symbolic forms.

17. A method of allocating resources according to claim 1, wherein the available resources are characterized by the following information:

- person identifier, person name, person type, shift assignment by day of week, task preference,
- shift name, shift start time, shift end time, lunch start, break 1 start, break 2 start,
- person type categories, eligible tasks,
- task name, rate per task, task capacity, task color for Gantt chart, flow percentages between tasks,
- projected incoming volume by task and time, and
- start of day queues in each task.

18. A method of allocating resources according to claim 1, wherein said assigning step (c) assigns the available resources using at least one of the following outputs:

- people allocation: number of people assigned to each task for each time period,
- volume data: number of RX's processed in each task for each time period,
- queue data: queue length for each task area by time period, and
- Gantt chart: person task assignment for each time period.

19. A method of allocating resources according to claim 1, wherein said assigning step (c), further comprises the steps of assigning the available resources to a varying set of tasks having varying individual rates.

20. A method of allocating resources according to claim 1, wherein said assigning step (c), further comprises the steps of assigning the available resources to the at least one task with a work flow between tasks following a Markov Chain.

21. A method of allocating resources according to claim 3, wherein the next event includes at least one of: a resource or task that becoming subsequently available, incoming work, a queue reaching zero, and a minimum time in the task.

22. A method of allocating resources according to claim 1, further comprising the step of repeatedly performing said steps (a) - (c) until the end of a predetermined time period is reached.

23. A method of allocating resources according to claim 1, further comprising the step performing the at least one task responsive to the resource assigned in said assigning step (c).

24. A method of allocating resources according to claim 1, wherein the work producing system comprises a pharmacy.

25. (Amended) A method of allocating resources including scheduling jobs from among a plurality of resources of a work-producing system, wherein the work producing system comprises a pharmacy, said method comprising the steps of:

- 5 (a) sorting, in a predetermined order, available resources to be utilized in the pharmacy by at least one of a task assignment, a number of tasks performable, rate per task, and cost per hour, and determining at least one queue responsive to said sorting, wherein the rate per task characterizes the units processed in the task for a time period;
- (b) normalizing the at least one queue by dividing a current task queue by an average rate of the available resources for each task in the current task queue; and
- 10 (c) assigning the available resources to at least one task with a predetermined normalized queue subject to at least one task constraint.

26. (Amended) A computer program memory, storing computer instructions to allocate resources including scheduling jobs from among a plurality of resources of a work-producing system, wherein the work producing system comprises a pharmacy, the computer instructions including:

- 5 (a) sorting, in a predetermined order, available resources to be utilized in the pharmacy by at least one of a last task assignment, a number of tasks performable, rate per task, and cost per hour, and determining at least one queue responsive to said sorting, wherein the rate per task characterizes the units processed in the task for a time period;
- (b) normalizing the at least one queue by dividing a current task queue by an
- 10 average rate of the available resources for each task in the current task queue; and
- (c) assigning the available resources to at least one task with a predetermined normalized queue subject to at least one task constraint.